Application of Optimization Techniques to a Nonlinear Problem of Communication Network Design With Nonlinear Constraints

Jeffrey E. Wieselthier, Gam D. Nguyen, Anthony Ephremides, and Craig M. Barnhart

***Abstract*—Nonlinear optimization under nonlinear constraints is usually difficult. However, standard ad-hoc search techniques may work successfully in some cases. Here, we consider an augmented Lagrangian formulation, and we develop a “projection heuristic” that “guides” the iterative search toward the optimum. We demonstrate the effectiveness of this approach by applying it to the problem of maximizing a circuit-switched communication network’s throughput under quality-of-service (QoS) constraints by means of choosing the input offered load. This problem is useful for “sizing” the network capacity. Performance results using several versions of the algorithm demonstrate its robustness, in terms of its accuracy and convergence properties.**

***Index Terms*—Admission control, communication network, optimization, performance evaluation, quality-of-service (QoS).**

I. Introduction

Nonlinear optimization problems with multiple nonlinear constraints are often difficult to solve, because although the available mathematical theory provides the basic principles for solution, it does not guarantee convergence to the optimal point [1]. The straightforward application of augmented Lagrangian techniques to such problems typically results in slow (or lack of) convergence, and often in failure to achieve the optimal solution. In this technical note, we introduce a “projection heuristic” that “guides” the iterative search more directly and more robustly to the optimal solution.

We illustrate the effectiveness of this heuristic by applying it to a problem that arises in communication networks, namely the maximization of throughput in multihop, circuit-switched networks that are subject to quality-of-service (QoS) constraints on blocking probability. The objective is to determine the offered-load profile that maximizes throughput, for specified routing and admission-control policies. This problem is useful for “sizing” the network capacity, i.e., for determining the maximum throughput that can be supported by the network, subject to QoS constraints [2]. Issues related to speed of convergence and quality of solution are addressed. Several versions of the algorithm are defined, and performance results are presented to illustrate their robustness.

II. The Optimization Problem

We are interested in nonlinear optimization problems with multiple nonlinear constraints. In this section, we use Lagrangian techniques to formulate the basic optimization problem.

1) *Constrained Optimization Problem:*

$$\max_{\lambda}\{S(\boldsymbol{\lambda})\} \tag{1}$$
$$\text{subject to: } P_j(\boldsymbol{\lambda}) \le Q_j \quad 1 \le j \le J$$

where $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_J)$ is a J-dimensional input vector, the performance measure $S(\boldsymbol{\lambda})$ is a nonlinear function of the input vector, and the Q_j are the values of the constraints imposed on nonlinear functions $P_j(\boldsymbol{\lambda})$ of the input vector. For example, in Section III we consider a circuit-switched networking example in which λ_j represents the offered load to circuit $j, S(\boldsymbol{\lambda})$ is throughput, and $P_j(\boldsymbol{\lambda})$ is the probability that an incoming call to circuit j is blocked.

Definitions:

- We say that an input vector $\boldsymbol{\lambda}$ is *admissible* if the constraints are satisfied.
- The *admissible region* contains all input vectors that are admissible.
- Corresponding to each admissible vector $\boldsymbol{\lambda}$ is a value of *admissible performance*.

We convert our constrained optimization problem to an unconstrained one by using the augmented Lagrangian function [1] given by

$$L(\boldsymbol{\lambda}, \gamma) = S(\boldsymbol{\lambda}) + \sum_{j=1}^{J}\left[\gamma_j \min\{0, Q_j - P_j(\boldsymbol{\lambda})\} - \frac{d}{2}(\min\{0, Q_j - P_j(\boldsymbol{\lambda})\})^2\right]. \tag{2}$$

Our goal is to maximize $L(\,\cdot\,)$ over $\boldsymbol{\lambda}$. To do this, we use the iterative procedure

$$\lambda_j(k+1) = \max\left\{\lambda_{\min}, \lambda_j(k) + \theta(k)\frac{\partial L(\boldsymbol{\lambda}, \gamma)}{\partial \lambda_j}\right\}$$
$$j = 1, \ldots, J \quad k = 1, \ldots, k_{\max} \quad \lambda_i(0) = \lambda_{io} \ge \lambda_{\min} \tag{3}$$

where $\theta(k)$ is a stepsize parameter, and

$$\frac{\partial L(\boldsymbol{\lambda}, \gamma)}{\partial \lambda_i} = \frac{\partial S}{\partial \lambda_i} + \sum_{j=1}^{J} 1(P_j(\boldsymbol{\lambda}) > Q_j) \times \frac{\partial P_j}{\partial \lambda_i}[d(Q_j - P_j(\boldsymbol{\lambda})) - \gamma_j]. \tag{4}$$

The Lagrange multipliers, γ_i, are updated according to

$$\gamma_j(k+1) = \gamma_j(k) - 1(P_j(\boldsymbol{\lambda}) > Q_j)\frac{c(Q_j - P_j(\boldsymbol{\lambda}))}{k} \tag{5}$$

where c is a positive constant and $\gamma_j(0) = \gamma_o, j = 1, \ldots, J$. The forms of the gradients of S and P_j are problem specific. A variety of rules we have used for updating the Lagrange multipliers and stepsize parameter are discussed in [2]. We refer to the straightforward application of the updating rule defined by (3), (4), and (5) as the “basic search technique.”

III. Motivation for This Formulation: A Networking Problem

We consider a circuit-switched network with predetermined paths between each pair of source and destination nodes throughout the duration of each accepted session (e.g., voice call). We assume the usual, “blocked calls cleared,” mode of operation, i.e., unless sessions are accepted for immediate transmission, they are “blocked” and lost from the system. Appropriate performance measures for this mode of operation include blocking probability and throughput.

Manuscript received September 29, 2000; revised June 29, 2001. Recommended by the Guest Editors W.-B. Gong and T. Başar. This work was supported by the Office of Naval Research.

J. E. Wieselthier and G. D. Nguyen are with the Code 5521, Information Technology Division, Naval Research Laboratory, Washington, DC 20375 USA (e-mail: wieselthier@itd.nrl.navy.mil; nguyen@itd.nrl.navy.mil).

A. Ephremides is with the Electrical and Computer Engineering Department and Institute for Systems Research, University of Maryland, College Park, MD 20742 USA (e-mail: tony@eng.umd.edu).

C. M. Barnhart was with the Naval Research Laboratory, Washington, DC 20375 USA. He is now with the TRW Data Technologies Division, Aurora, CO 80011 USA (e-mail: Craig.Barnhart@trw.com).

Publisher Item Identifier S 0018-9286(02)05509-5.

We consider J source-destination pairs, each of which is assigned a fixed multihop path (circuit) through the graph of the network that interconnects them. We let x_j (which may be greater than 1) denote the number of sessions that are ongoing on the jth such circuit, and we assume that each accepted session consumes a fixed amount of resource throughout its duration, i.e., a fixed unit of bandwidth is required over each link in the circuit to support each session. The state of the system is the J-dimensional vector $\mathbf{x} = (x_1, \ldots, x_J)$.

The capacity of network element (link or node) i is denoted by T_i. In the wired case, T_i is the number of channels supported by link $i, i = 1, \ldots, M$, where M is the number of links in the network. In the wireless case, T_i is the number of transceivers at node i, and M is the number of nodes in the network.[1] Each network element can support sessions corresponding to several circuits simultaneously, as long as the state variables $x_1, x_2, \ldots, x_J$ satisfy sets of linear constraints of the form

$$\sum_{j \in I_i} x_j \le T_i, \qquad i = 1, \ldots, M \tag{6}$$

where I_i is the set of circuits that share network element i.

A. Admission Control

Our ultimate goal is to achieve optimal network performance, which, however, depends on a large number of factors, notably routing, admission control, and offered traffic. In [3] and [4], we approached this problem by exercising an admission-control policy on calls, under the assumption that routes and offered loads on each of the circuits were fixed. In this note, we again fix the routes, but instead of determining the best admission-control policy for a fixed offered load, we determine the offered load that maximizes throughput for a fixed admission-control policy, subject to QoS constraints on blocking probability.

We restrict our admission control policies to the class of "threshold" policies. Threshold controls restrict the number of calls that will be admitted to the individual circuits, and can be expressed as

$$x_j \le X_j, \quad 1 \le j \le J \tag{7}$$

where X_j is the threshold on circuit j. Transceivers are not assigned a priori to circuits; sessions are accepted as long as the threshold values (the X_j's) are not exceeded. In [3] and [4], we also studied "linear-combination" controls.

Policies that use threshold and/or linear-combination controls are a subclass of the "coordinate-convex" policies [5]. A stationary admission-control policy is specified in terms of the set of allowable states Ω. A new call is admitted if the state to be entered is in the allowable region; otherwise, it is blocked and lost from the system. Coordinate-convex control policies are used because they provide a form of intelligent resource sharing without the complexity of dynamic programming.

B. The Solution and Performance Measures

We assume Poisson arrival statistics, and denote the offered load vector by $\boldsymbol{\lambda} = (\lambda_1, \lambda_2, \ldots, \lambda_J)$, where λ_j is the arrival rate to circuit j. The service rate vector is $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_J)$, where we let $\mu_j = 1\ (1 \le j \le J)$. Thus, the corresponding offered load on circuit j is $\rho_j = \lambda_j$. Furthermore, control is centralized, and the resources needed to support a circuit are acquired simultaneously when the call arrives and are released simultaneously when the call is completed. Calls are blocked when one or more nodes along the path do not have a transceiver available or when a decision is made not to accept a call, i.e., to accept the call would bring the system state outside the region defined by admission-control policy Ω. Under these conditions, in conjunction with the use of coordinate-convex policies, it has been shown [6], [7] that the system state has the product-form stationary distribution.[2] For any allowable state space Ω, it is straightforward (though time consuming) to evaluate the normalization constant, which in turn permits the evaluation of performance measures such as throughput and blocking probability, which we define as follows:

$$\begin{aligned} Sj(\boldsymbol{\lambda}) &= \text{throughput on circuit } j \\ &= \lambda_j(1 - P_j(\boldsymbol{\lambda})) \end{aligned} \tag{8}$$

$$\begin{aligned} S(\boldsymbol{\lambda}) &= \text{total throughput} = \sum_{j=1}^{J} S_j(\boldsymbol{\lambda}) \\ &= \Lambda(1 - P_{\mathrm{av}}(\boldsymbol{\lambda})) \end{aligned} \tag{9}$$

$$\begin{aligned} P_{\mathrm{av}}(\boldsymbol{\lambda}) &= \text{overall blocking probability} \\ &= \sum_{j=1}^{J} \frac{\lambda_j}{\Lambda} P_j(\boldsymbol{\lambda}). \end{aligned} \tag{10}$$

where $P_j(\boldsymbol{\lambda})$ is the probability that an incoming call to circuit j is blocked and $\Lambda = \sum_{j=1}^{J} \lambda_j$ is the overall arrival rate.

The circuit blocking probabilities $P_i(\boldsymbol{\lambda})$, the circuit throughput values $S_i(\lambda)$, and the partial derivatives (gradients) $\partial P_j(\boldsymbol{\lambda})/\partial\lambda_i$ and $\partial S_j(\boldsymbol{\lambda})/\partial\lambda_i$, which are used in the Lagrangian update equation (4), are obtained from the product-form solution. In [6], Jordan and Varaiya showed that

$$\frac{\partial P_j(\boldsymbol{\lambda})}{\partial \lambda_i} = \begin{cases} -\frac{\mu_j}{\lambda_i \lambda_j} \mathrm{cov}(x_i, x_j), & i \ne j \\ \frac{\mu_i}{\lambda_i^2}(E\{x_i\} - \mathrm{var}(x_i)), & i = j \end{cases} \quad \text{and}$$

$$\frac{\partial S_j(\boldsymbol{\lambda})}{\partial \lambda_i} = \frac{\mu_j}{\lambda_i} \mathrm{cov}(x_i, x_j). \tag{11}$$

IV. Guided Search Techniques

When the basic search technique is applied to the networking problem of Section III, significant (although nonmonotonic) progress is typically made in the early stage of the search, whereas considerably less-productive oscillatory behavior is observed as the search progresses. Moreover, the quality of the solution is often sensitive to the starting point of the search. A common difficulty in constrained optimization problems arises because the optimum lies on the search boundary (i.e., one or more of the circuit blocking probabilities is at the maximum-permitted QoS value). In unconstrained optimization problems, gradient search procedures are naturally slowed (smaller steps) by the decreasing gradient as they approach the optimum. This slowing allows the search to ascend smoothly to the maximum. However, when the optimum lies on the boundary, as it often does in constrained problems,[3] there is not necessarily a decrease in the gradient in its neighborhood. In this case, typical gradient search techniques rely on damping of the stepsize θ to cause the search to slow and home in on the optimum. However, an overly rapid decrease in θ results in failure to reach the optimal solution, whereas a less rapid decrease in θ can result in unacceptably slow convergence.

[1] For a wireless network model, this translates to the assumption that each node has several transceivers, and that each session requires the use of one transceiver at every node in its path; FDMA can then be conveniently assumed for channel access, provided that there is sufficient bandwidth for all transceivers to operate simultaneously at noninterfering frequencies.

[2] It is not necessary to assume that the call duration is exponential. A Poisson arrival process and general service time distribution is sufficient for the product-form solution to apply [8]; knowledge of the means of the service times provides enough information to determine the equilibrium distribution.

[3] We believe that, in our optimization problem, at least one of the circuit blocking probabilities at the optimal point must be at the maximum permitted value. This conjecture is supported by extensive empirical evidence in a variety of network examples. We have observed that, typically, between half and all of the circuit blocking probabilities are at the maximum permitted value.

The most interesting, and troublesome, behavior occurs when the search trajectory passes near the QoS constraint contour. The violation of a constraint results in oscillatory behavior with little progress toward the optimal point. The desired behavior would be for the search to proceed along the contour corresponding to the QoS constraint, rather than at a significant angle to it. We have attempted to mitigate the oscillatory behavior of the basic search technique by using our knowledge of the throughput and blocking probability gradients to guide the search more efficiently.

A. Guiding the Search: Preliminary Approach

To illustrate the principle of guided search, we consider an example in which the blocking probability of the "dominant circuit" (i.e., the circuit with the largest blocking probability) is close to (say within some ε of) the specified QoS value. We would like to guide the search in a direction of increasing throughput, so that it tends to proceed parallel to the contour at which the blocking probability is at the specified QoS value. To simplify the discussion, let us first consider the case in which exactly one of the circuit blocking probabilities (the dominant circuit) is located within ε of the QoS constraint, i.e., $Q_j - \varepsilon \le P_j(\boldsymbol{\lambda}) \le Q_j + \varepsilon$, for exactly one value of $j \in \{1, 2, \ldots, J\}$. Let us call this circuit c. In this case, we would like the search to proceed along the component of the throughput gradient ∇S that is orthogonal to the circuit blocking probability gradient ∇P_c at our current point in the search. By eliminating the component parallel to ∇P_c, we discourage increase in the blocking probability of the dominant circuit. The desired projection can be written as

$$\{\text{Component of } \nabla S \text{ orthogonal to } \nabla P_c\} = \nabla S - \frac{\nabla S \cdot \nabla P_c}{\|\nabla P_c\|^2} \nabla P_c \qquad (12)$$

where

$$\nabla S \cdot \nabla P_c = \sum_{i=1}^{J} \frac{\partial S}{\partial \lambda_i} \frac{\partial P_c}{\partial \lambda_i} = \sum_{i=1}^{J} \sum_{j=1}^{J} \frac{\partial S_j}{\partial \lambda_i} \frac{\partial P_c}{\partial \lambda_i} \qquad (13)$$

and $\|\boldsymbol{X}\| = \sqrt{\sum_{j=1}^{J} X_j^2}$ is the norm of the vector $\boldsymbol{X}$. Then we introduce a vector $\boldsymbol{D} = (D_1, D_2, \ldots, D_J)$ (see Fig. 1), which is equal to this projection when the blocking probability of the dominant circuit is located in a band of width 2ε centered about the QoS contour; otherwise, $\boldsymbol{D}$ is equal to the throughput gradient ∇S

$$\boldsymbol{D} = \begin{cases} \nabla S - \dfrac{\nabla S \cdot \nabla P_c}{\|\nabla P_c\|^2} \nabla P_c, & \text{QoS} - \varepsilon \le P_c \le \text{QoS} + \varepsilon \\ \nabla S, & \text{otherwise.} \end{cases} \qquad (14)$$

We modify the Lagrangian objective function of (4) by inserting D_i in place of $\partial S / \partial \lambda_i$ as follows:

$$\frac{\partial L(\boldsymbol{\lambda}, \boldsymbol{\gamma})}{\partial \lambda_i} = D_i + \sum_{j=1}^{J} 1(P_j(\boldsymbol{\lambda}) > Q_j) \times \frac{\partial P_j}{\partial \lambda_i} [d(Q_j - P_j(\boldsymbol{\lambda})) - \gamma_j]. \qquad (15)$$

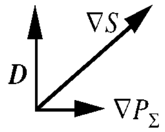


Fig. 1. $\boldsymbol{D} =$ component of ∇S that is orthogonal to ∇P_{Σ}.

B. Guiding the Search: Generalized Approach

The use of the projection operation described above removes the component of ∇S that is parallel to ∇P_c. By doing so, we update $\boldsymbol{\lambda}$ in a direction that increases throughput without increasing P_c. However, the typical consequence of doing so is that one or more of the other circuits will soon violate the QoS constraint. At a typical point in the search, it is common for several circuits to violate the QoS constraint or to be sufficiently close to the QoS boundary that the QoS constraint is in danger of being violated. For example, we have observed behavior in which the chosen circuit for the projection alternates among two or three of the circuits, resulting in oscillatory behavior in which little progress is made toward the optimal solution. To mitigate this behavior, we have considered a generalized form of the projection operation in which several circuits are included in the projection. The inclusion of several circuits takes into consideration the fact that we are dealing with a number of constraints simultaneously. Thus, we would like to update $\boldsymbol{\lambda}$ in a direction that discourages violation of any of the QoS constraints.

To incorporate the QoS constraints associated with some or all of the circuits into the search-guiding mechanism, we introduce the quantity P_{Σ}, which is a function of the circuit-blocking probabilities $P_1, P_2, \ldots, P_J$. In this note, we have used the following simple, linear form for P_{Σ}:

$$P_{\Sigma} = \sum_{i \in \Sigma} P_i \qquad (16)$$

where Σ is a subset of $\{1, 2, \ldots, J\}$. The vector $\boldsymbol{D}$, introduced in (14), is then rewritten as (17), as shown at the bottom of the page. The projection vector $\boldsymbol{D}$ specified by (17) removes the component of ∇S that is in the direction of the gradient of the *average* blocking probability of the circuits included in Σ. This expression is identical to that of (14), except that P_{Σ} replaces P_c in the dot products, and that the projection operation is used only when the resulting value of $\|\boldsymbol{D}\|$ is sufficiently large. The reason for using the projection operation only when it provides a sufficiently large value of $\|\boldsymbol{D}\|$ is based on our experimental observation that (in some cases) the trajectory can reach a point at which $\|\boldsymbol{D}\|$ is quite small. This behavior results in slow progress toward the optimal point, or even virtually total stopping of the trajectory, resulting in premature convergence; in fact, the trajectory can converge to a point interior to the admissible region (thus none of the circuit blocking probabilities are at the specified value, a condition not characteristic of the optimal point). This behavior is especially prevalent when the set Σ is large (e.g., we have considered cases in which Σ contains all J circuits). It occurs when the gradients of S and P_{Σ} are nearly parallel to each other. Turning off the projection operation (typically for just a single iteration) permits the trajectory to escape from such undesirable points. We have found that a value of $\tau = 0.1$ works well.

Care must also be taken in the choice of several other parameters used in the algorithm, such as the choice of c (used in updating the Lagrange multipliers in (5) and d (which weights the penalty term in

$$\boldsymbol{D} = \begin{cases} \nabla S - \dfrac{\nabla S \cdot \nabla P_{\Sigma}}{\|\nabla P_{\Sigma}\|^2} \nabla P_{\Sigma}, & \text{if } \left\| \nabla S - \dfrac{\nabla S \cdot \nabla P_{\Sigma}}{\|\nabla P_{\Sigma}\|^2} \nabla P_{\Sigma} \right\| > \tau \|\nabla S\| \\ \nabla S, & \text{otherwise} \end{cases}. \qquad (17)$$

(15). The use of $c = d = 50$ worked well for high values of QoS (e.g., ≥ 0.2), but not for more realistic values. The relatively poor performance for low values of QoS was observed because the gradient terms $\partial P_j/\partial \lambda_i$ were too small to drive the search back into the admissible region at the low offered loads that are characteristic of low values of QoS. We have observed experimentally that this problem can been mitigated by weighting the constraint-violation terms by $1/\sqrt{Q_j}$ (while maintaining $c = d = 50$) as follows:

$$\frac{\partial L(\boldsymbol{\lambda}, \boldsymbol{\gamma})}{\partial \lambda_i} = D_i + \alpha \sum_{j=1}^{J} 1(P_j(\boldsymbol{\lambda}) > Q_j) \times \frac{\partial P_j}{\partial \lambda_i} \frac{[d(Q_j - P_j(\boldsymbol{\lambda})) - \gamma_j]}{\sqrt{Q_j}} \quad (18)$$

where α is a "kick-up" factor that can be updated (increased from an initial value of 1) as necessary, e.g., α can be increased if too many consecutive inadmissible solutions are observed, or decreased if too many consecutive admissible solutions are observed (after the inadmissible region has been entered at least once). The incorporation of these heuristic fixes into the update equation has resulted in a robust algorithm that does not require the fine-tuning of parameters.

V. Alternative Versions of the Algorithm

We have studied several versions of the algorithm based on (18), which differ in their use of the dot-product projection and in the stepsize update rule. Here, we briefly describe one of our approaches; a complete discussion is provided in [2]. In our discussion, it is implicitly assumed that $Q_j = \text{constant}, j = 1, \ldots, J$, (i.e., that all circuits are subject to the same constraint on maximum blocking probability), although it is certainly possible to define projection rules that incorporate different QoS values (see [2]).

A. Projection Rules

The projection rule, as described in Section IV-B, guides the search by removing the component of the throughput gradient that is parallel to ∇P_Σ, where $P_\Sigma = \sum_{j \in \Sigma} P_j$, for some subset Σ of $\{1, 2, \ldots, J\}$. The effect of the projection is to remove the component of the throughput gradient that is in the direction of the gradient of the sum of the blocking probabilities (or, equivalently, the gradient of the average blocking probability) of the circuits included in Σ. By including several circuits in Σ, it is possible to discourage (although not necessarily prevent) the blocking probabilities of these circuits from exceeding the QoS constraint value. In addition, the oscillatory behavior that results from the use of a single circuit (the identity of which typically alternates among a small set of circuits) in the dot product is reduced. However, it must be acknowledged that the use of the projection is a heuristic approach. The performance results presented in Section VI and [2] demonstrate that, if used judiciously, the projection can, in fact, be very helpful.

In most versions of the algorithm studied in the core runs of [2], we used a version of the projection rule in which Σ is defined as follows:

$$\Sigma = \{j : P_j \geq p_{\min} + \nu(p_{\max} - p_{\min})\} \quad (19)$$

where $p_{\min} = \min\{P_j, j = 1, 2, \ldots, J\}$, $p_{\max} = \max\{P_j, j = 1, 2, \ldots, J\}$, and $\nu \in [0, 1]$. The parameter ν can be chosen to include either few or many circuits, as desired. For example, for the network discussed in this note, the choice of $\nu = 0.2$ causes, on the average, about eight (out of ten) circuits to be included in Σ (thus Σ is a large set). Alternative choices for the set Σ are considered in [2].

We have observed that the use of a large set Σ tends to keep the trajectory well inside the admissible region during the early phase of the search, and discourages the trajectory from straying too far into the inadmissible region once the QoS-constraint boundary has been crossed. However, although the neighborhood of the optimal point is reached rapidly, it is common for the trajectory to proceed past it, eventually converging to a point relatively far from the optimal. Apparently, the algorithm does not converge to the true optimal point because of the distortion introduced by the use of $\boldsymbol{D}$ rather than ∇S.

Based on these observations, which have been supported by extensive numerical results, we have concluded that it is often best to use a large set Σ during the early phase of the search, and then to turn off the projection term (i.e., set $\Sigma = \emptyset$, the empty set) at some point during the search. When the projection is turned off, the final approach to the optimal solution can be made without the presence of distortion.

B. Stepsize Considerations

Typically, we have chosen the initial stepsize θ_0 on the basis of a short pilot run in which the projection is not used; it is chosen so that, starting at $\lambda_i = 0$, the trajectory exits the admissible region for the first time after about five to fifteen iterations. The same value of θ_0 is used whether or not the projection is used in the actual search.

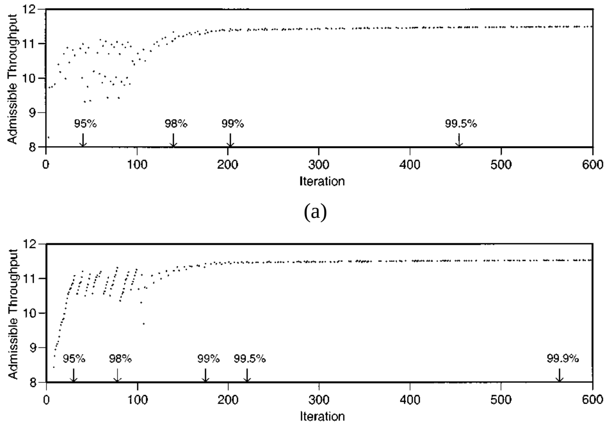
We have found that a first exit point of five iterations works well for large values of the QoS constraint, e.g., 0.3. However, this approach appears to produce an excessively large initial stepsize for small values, e.g., 0.001. Thus, in some of our examples for $Q_j = 0.001$ we have used an initial value of θ that is half that produced by using the rule based on exiting the admissible region for the first time at the fifth iteration. To explain the difference in behavior, consider the terms $\partial S_i/\partial \lambda_i$ derived from (4), which are usually significantly larger than the terms $\partial S_i/\partial \lambda_j$, when $j \neq i$. These "diagonal" terms have a value close to 1 at the low offered loads that are characteristic of low blocking probability; however, these terms are considerably smaller at offered loads characteristic of significantly higher blocking probability (typical average values are approximately 0.3 in many of our examples for $Q_j = 0.3$). Thus the use of smaller stepsizes at low QoS values compensates for the larger values of throughput gradient at the corresponding offered loads.

VI. Performance Results for a Networking Example

In this section, we discuss the performance of the search algorithms in terms of the evolution of the admissible throughput as the search progresses. We refer to the version that uses the projection in the first phase, simply as the "projection algorithm." We also present results for the basic search technique, which does not use the projection at all. Both of these versions are based on a stepsize rule in which θ is constant for 100 iterations, then decreases exponentially to 0.1 of its initial value after an additional 100 iterations, and then decreases exponentially to 0.001 of its initial value after an additional 800 iterations. In the version with the projection algorithm, $\nu = 0.2$ is applied for the first 100 iterations; the projection operation is turned off by setting $\Sigma = \emptyset$ for the next 900 iterations. Alternative stepsize and projection rules are discussed in [2].

Fig. 2 shows the "admissible" throughput (i.e., values are not shown when the QoS constraint is violated) for both the basic search technique and projection algorithm for the case of Network 1 with $T_i = 6, X_j = 4$, and $Q_j = 0.3 (1 \leq j \leq J)$.[4] This example is typical, in

[4] This "unrealistically high" value of blocking probability was used because it typically results in a more-difficult optimization problem than lower values (e.g., $Q_j = 0.001$), in the sense that a greater number of iterations is usually needed.



(a)

(b)

Fig. 2. Evolution of admissible throughput; $Q_j = 0.3$. (a) Basic search technique. (b) Projection algorithm.

that the use of the projection operation provides a smoother ascent to good throughput values early in the search, and hence typically faster attainment of the 95% and 98% milestones[5] [2]. However, there is usually less difference in the speed with which the higher milestones are reached, and sometimes the basic search technique reaches them faster.

We discovered in our early studies that the use of the projection operation often prevents convergence to the optimal solution, especially when a relatively large number of circuits are included in the projection. We may view the use of the projection term in the first phase(s) as the determination of an "initial condition" for the "undistorted" version of the algorithm (i.e., the version without the projection term). Thus, as long as the trajectory is brought sufficiently close to the neighborhood of the optimal solution in the early phase(s), the undistorted version of the algorithm should bring the solution close to the optimal point before the end of the allotted 1000 iterations.

Even for network examples in which all versions converge to nearly the same point, the use of the projection operation can have a profound impact on the behavior of the algorithm. For example, when a large number of circuits are included in the projection set Σ (e.g., by using a relatively small value of ν such as 0.2), a relatively smooth (although perhaps somewhat slow) trajectory is observed in which the throughput increases monotonically to a large percentage of the benchmark throughput value before exiting the admissible region for the first time. By contrast, when the projection operation is not used, the trajectory is much rougher, with considerably larger deviations in offered load and throughput from one iteration to the next. Although it is indeed possible to achieve some of the high milestone values relatively early in the run when the projection is not used, it may be a matter of "luck" as to whether or not such points are indeed found early. Even if they are found, the trajectory will often move far from these points because of the large stepsize. Based on the extensive testing discussed in [2], it appears that the smoothing effect of the projection operation with $\nu = 0.2$ permits the effective use of relatively aggressive stepsize rules, thus permitting faster convergence.

Our primary conclusion, obtained by examining the data presented in [2], is that virtually all versions of the algorithm perform well, based on the criterion of providing optimal (or nearly optimal) throughput within 1000 iterations. However, use of the projection operation in the early part of the search can be beneficial. For example, it typically results in reaching the 95% and 98% milestones faster than is possible with the basic search technique, and, as just noted it permits the use of more aggressive stepsize rules, which result in faster overall convergence.

A. An Observation

One characteristic property of the optimal solution in constrained optimization problems such as ours is that at least one of the circuit blocking probabilities must be at the maximum permissible value, i.e., at Q_j. To measure how close the individual circuits approach this value, we introduce the normalized circuit blocking probabilities

$$\hat{P}_j = P_j/Q_j, \qquad j = 1, \ldots, J. \tag{20}$$

Thus $\hat{P}_j = 1$ when $P_j = Q_j$.

The fact that not all blocking probabilities are near the specified QoS level when $Q_j = 0.3$ is not surprising. It is not a failure of the algorithm, but rather reflects the fact that the level of interaction among the circuits increases as offered load increases.[6] Thus, there does not exist a set of offered-load values for which all blocking probabilities are at the maximum permitted QoS value when that value is relatively high (e.g., 0.3).

On the basis of these observations, as well as additional discussion in [2], it appears that whenever the optimal solution does, in fact, lie very close to the QoS contour in all dimensions, there is very little difference in the quality of the solutions produced by the various versions of the algorithm. Also, it appears that our algorithm is more robust in such cases; typically, fewer iterations are needed, and more aggressive stepsize rules (resulting in faster attainment of milestones) are usually successful. Furthermore, we believe that one can have more confidence in the quality of the solution if the blocking probabilities are all close to the QoS constraint value. In some cases (particularly when several of the blocking probabilities are far from the QoS boundary), the network designer/manager might want to run several versions of the algorithm to ensure that the solution is close to the true optimum.

B. An Alternative QoS Constraint: Average Blocking Probability

In [2], we also considered an alternative version of the QoS constraint in which we require only that the average blocking probability in the network satisfy this constraint. It was shown that relaxing the QoS constraint in this manner results in not only higher throughput values, but also in considerably faster convergence, even when the projection operation is not used. Both of these characteristics are a consequence of the need to satisfy only a single average QoS constraint, which permits the set of offered loads to trade off among themselves more easily than the case in which the QoS constraint must be satisfied on each individual circuit. In view of the ability of the basic search technique to obtain optimal solutions rapidly and reliably without using the projection operation, we do not consider this alternative constraint in the present note.

VII. Conclusion

In this note, we have addressed the solution of nonlinear optimization problems with multiple nonlinear constraints, based on the use of Lagrangian techniques with a penalty function. We observed several shortcomings associated with standard Lagrangian techniques. First, there was no guarantee of convergence and no guarantee of approaching the optimal solution. Second, there were many parameters that could be "tuned" and thus affect the solution. Third, the direct use of standard versions of the Lagrangian techniques were very slow and often fraught with oscillations.

[5] For example, the "95% milestone" is the first point at which an admissible throughput value as high as 95% of the best value (observed for any algorithm for the current problem) is obtained.

[6] The values of the partial derivatives $(\partial P_i/\partial\lambda_j)$ used in the update equations are increasing functions of the offered load.

Therefore, we proposed a heuristic modification to the search algorithm, which is based on the use of the projection of the gradient on an appropriate plane determined by the constraint surfaces, and found that there was improvement in all aspects of the search. If, in addition, fine tuning of the parameters was used, the resulting results were indicative (although, still, not assuring) of convergence to the optimal solution at reasonable speeds.

In this note we have applied the projection algorithm to a nonstandard problem in communication networking. The proposed problem is useful and meaningful in two distinct ways. First, it establishes a "capacity-like" result for a given network in which the routes are fixed. In other words, even though the network operator normally will not choose the input load vector (although via pricing controls even this choice can be implemented), it will be possible to predetermine what the ultimate capabilities of the network are for the chosen set of routes. That is, it will permit the network operator to "size" the network and thus enrich the control capabilities in its operation.

Second, the optimal routing problem, i.e., finding the best routes for a given input load, although a typical network operation problem, is essentially unsolvable. It is an NP-complete combinatorial optimization problem. This is why routing in circuit-switched networks, like the Public Switched Telephone Network, has been the object of study for many years and has generated a large number of suboptimal heuristics. This is in contrast to the packet-switched, datagram routing problem, which is a well-behaved and essentially solved problem. Therefore, when a set of routes is chosen for a given input load, it is likely to be used for a period of time even if the input load changes. Dynamic adjustment of heuristically obtained suboptimal routes on a short time scale is not feasible, nor does it make much sense. Consequently, the approach we introduce in this note permits the network operator to establish the maximum throughput this set of routes is capable of carrying (while meeting the blocking probability requirements), and thereby establish how much of a gap there is between the achieved throughput and the achievable throughput (i.e., how much of a mismatch there is between the actual input load and the actual set of routes). This knowledge could be used, in fact, as a criterion for deciding whether to re-solve the routing problem and change the set of circuit paths of the network. Thus, although on its surface the problem we propose may appear unorthodox, we believe it offers a totally novel tool for network operation and design.

Although we did not investigate the applicability and usefulness of this heuristic in other nonlinear optimization problems (from the networking area or from other disciplines), we suspect that it possesses inherent robustness properties that are likely to make it applicable elsewhere as well. We also believe that our investigation yields further evidence that, in the field of communication networks, there are opportunities for fertile use of optimization theory techniques, as observed in [9].

References

[1] D. P. Bertsekas, *Constrained Optimization and Lagrange Multiplier Methods*. Belmont, MA: Athena Scientific, 1996.

[2] J. E. Wieselthier, G. D. Nguyen, and A. Ephremides, "Throughput maximization under quality of service constraints," Naval Research Laboratory, NRL Formal Rep. 5520-00-9922, June 2000.

[3] C. M. Barnhart, J. E. Wieselthier, and A. Ephremides, "Admission-control policies for multihop wireless networks," *Wireless Networks*, vol. 1, no. 4, pp. 373–387, Dec. 1995.

[4] ——, "Admission control in integrated voice/data multihop radio networks," Naval Research Laboratory, Rep. NRL/MR/5521-93-7196, Jan. 18, 1993.

[5] J. M. Aein, "A multi-user-class, blocked-calls-cleared, demand access model," *IEEE Trans. Commun.*, vol. COM-26, pp. 378–385, Mar. 1978.

[6] S. Jordan and P. Varaiya, "Throughput in multiple service, multiple resource communication networks," *IEEE Trans. Commun.*, vol. 39, pp. 1216–1222, Aug. 1991.

[7] ——, "Control of multiple service, multiple resource communication networks," *IEEE Trans. Commun.*, vol. 42, pp. 2979–2988, Nov. 1994.

[8] D. Y. Burman, J. P. Lehoczky, and Y. Lim, "Insensitivity of blocking probabilities in a circuit-switching network," *J. Appl. Probab.*, vol. 21, pp. 850–859, 1984.

[9] A. Ephremides and S. Verdu, "Control and optimization methods in communication network problems," *IEEE Trans. Automat. Contr.*, vol. 34, pp. 930–942, Sept. 1989.